

⑬ 日本国特許庁(JP)

⑭ 特許出願公開

⑫ 公開特許公報(A)

昭61-837

⑮ Int. Cl.⁴

識別記号

庁内整理番号

⑯ 公開 昭和61年(1986)1月6日

G 06 F 9/06
12/02

B-7361-5B
6974-5B

審査請求 未請求 発明の数 1 (全5頁)

⑰ 発明の名称 プログラム・オーバレイロード方式

⑱ 特 願 昭59-120004

⑲ 出 願 昭59(1984)6月13日

⑳ 発 明 者 東 田 隆 司 尾張旭市晴丘町池上1番地 株式会社日立製作所旭工場内
㉑ 発 明 者 露 木 陽 介 尾張旭市晴丘町池上1番地 株式会社日立製作所旭工場内
㉒ 出 願 人 株式会社日立製作所 東京都千代田区神田駿河台4丁目6番地
㉓ 代 理 人 弁理士 高橋 明夫 外1名

明 細 書

1. 発明の名称

プログラム・オーバレイロード方式

2. 特許請求の範囲

(1) ルート・モジュールとプログラム・ロードにより外部ファイルから要求されたプログラム・モジュールを読み出し、該プログラム・モジュールを内部メモリのユーザ・プログラム・エリアにオーバレイロードさせる方式において、上記内部メモリにユーザ・プログラム・エリアとは別個にプログラム常驻化バッファ・エリアを設け、上記ルート・モジュールが優先指定情報を付して上記プログラム・ロードにロード要求を行うことにより該プログラム・ロードは優先指定がされたプログラム・モジュールをプログラム常驻化バッファ・エリアからユーザ・プログラム・エリアにロードし、該プログラム常驻化バッファ・エリアに上記プログラム・モジュールがないときには、該プログラム・モジュールを外部ファイルから上記プロ

グラム常驻化バッファ・エリアにロードしてモジュール名をインデックスに登録した後、該プログラム・モジュールを上記ユーザ・プログラム・エリアにロードすることを特徴とするプログラム・オーバレイロード方式。

3. 発明の詳細な説明

(発明の利用分野)

本発明は、プログラム・オーバレイロード方式に関し、特に使用頻度の高いプログラム・モジュールのロード時間を短縮でき、かつ有限のメモリ・プログラム・エリアを有効に活用できるプログラム・オーバレイロード方式に関するものである。

(発明の背景)

メモリ・アドレッシングとして限定された範囲しか動作しないような言語プログラムでは、オーバレイ・プログラム構造を用いることにより機能を多く持たせることができる。この場合には、システムにメモリの余裕があつても、オーバレイ構造を避けることはできず、オーバレイロードの時間がシステムの性能に影響を及ぼしている。勿論

システムにメモリの余裕がない場合には、ユーザ・プログラムをロードする際に、オーバレイロード方式を用いなければならない。

一般に、ユーザ・プログラム・エリアはある範囲内に限定されているため、外部ファイルに格納されているプログラムの全容量はそのエリアを越えてしまう。その場合には、メモリ・エリアを有効に使うため、あるプログラムをロードして実行した後、もう使用しなくなつたとき、その上から次のプログラムをロードする方法、すなわちオーバレイロードによる方法が行われる。しかし、この方法では、プログラム・ローダが必要の都度、外部ファイルからメモリ・エリアにオーバレイロードするため時間がかかり、きわめて効率が悪い。また、頻繁に使用するプログラムのみでオーバレイロードを行う場合には、一旦消されたプログラムを繰り返しロードしなければならない、さらに効率が悪くなる。頻繁に使用するプログラムは、メモリ上に常駐化しておくことが望ましいが、メモリ容量の制限により、一定のオーバレイロードを

実行した後、常駐化メモリ容量を越えた分のロードは、必ず外部ファイルからのオーバレイロードをすることになるため、使用頻度の高いプログラム・モジュールを選択して常駐化することができない。

(発明の目的)

本発明の目的は、このような欠点を解消し、使用頻度の高いプログラム・モジュールのロード時間を短縮でき、かつ有限のメモリ・エリアを最大限に有効利用できるプログラム・オーバレイロード方式を提供することにある。

(発明の概要)

上記目的を達成するため、本発明のプログラム・オーバレイロード方式は、内部メモリにユーザ・プログラム・エリアとは別個にプログラム常駐化バッファ・エリアを設け、ルート・モジュールが優先指定情報を付してプログラム・ローダにロード要求を行うことにより、該プログラム・ローダは優先指定情報が付されたプログラム・モジュールをプログラム常駐化バッファ・エリアからユー

ザ・プログラム・エリアにロードし、該プログラム常駐化バッファ・エリアに上記プログラム・モジュールがないときには、該プログラム・モジュールを外部ファイルから上記プログラム常駐化バッファ・エリアにロードしてモジュール名をインデックスに登録した後、該プログラム・モジュールを上記ユーザ・プログラム・エリアにロードすることに特徴がある。

(発明の実施例)

以下、本発明の実施例を、図面により説明する。

第1図は、本発明の一実施例を示すシステム・ブロックおよびメモリ・マップの図である。

第1図において、1はプログラム・モジュールを格納する外部ファイルであり、具体的にはフレキシブル・ディスク装置または固定ディスク装置である。2は主メモリであり、主メモリ2上にはOS (Operating System) 格納エリアと、ユーザ・プログラム・エリア5と、プログラム常駐化バッファ・エリア8が設定されている。3は主メモリ2上にロードされたあるいは外部ファイル1上に

格納されたプログラム・モジュール、10はデータ・バス、11はCPU、12は出力装置であるCRTディスプレイ装置、13は入力装置であるキーボードである。CRTディスプレイ装置12とキーボード13は、データ・バス10を介してCPU11と接続されており、オペレータのコンソールとして動作する。主メモリ2のOS格納エリア3には、OS部を構成するOSモジュール群14と、これらモジュール群14を制御し、かつオペレータとの交信を行うモニタ15が格納されている。なお、OSモジュール群14の中には、ユーザ・プログラム常駐部であるルート・モジュール6と、ユーザ・プログラム非常駐部である複数本のプログラム・モジュール9とを、ユーザ業務処理プログラムが実行されるユーザ・プログラム・エリア5にロードするプログラム・ローダ4が存在する。ルート・モジュール6は、オペレータからのプログラム・ロードコマンドにより、モニタ15を介してプログラム・ローダ4によりユーザ・プログラム・エリア5に業務処理プログラ

ムとしてロードされる。そして、ルート・モジュール6は、一旦ロードされた後は、ユーザ・プログラム・エリア5に常駐化され、以降、プログラム・ローダ4を起動してプログラム・モジュール9をユーザ・プログラム・エリア5にロードし、プログラム・モジュール9とともにユーザ業務処理を行う。

第1図における主メモリ2には、新たに複製のプログラム・モジュール9を常駐化しておくためのプログラム常駐化バッファ8を設け、このプログラム常駐化バッファ8にどのプログラム・モジュール9がロードされているかを示すインデックス(INDEX)部7も常駐させる。インデックス部7には、プログラム常駐化バッファ8にロードされたプログラム・モジュール9の名称が登録される。

このような主メモリ2のプログラム格納エリアマップにしたがつて、外部ファイル1に格納されているプログラム・モジュール9がデーモ・バス10を介してCPU11の制御によりプログラム

常駐化バッファ8またはユーザ・プログラム・エリア5にロードされる。

第1図の実施例においては、上述のように、主メモリ2上のプログラム常駐化バッファ8に複製個分のプログラム・モジュール格納エリアとインデックス部のエリアを用途し、先ずルート・モジュール6が優先指定情報を付してプログラム・ローダ4にプログラム・オーバレイロードを要求する。ここで、優先指定情報が付されるプログラムとしては、使用頻度が大でかつ重要なものが選択される。なお、優先指定情報が付されるプログラムがプログラム常駐化バッファ8に全部ロードできれば最も望ましいが、常駐化バッファ8の容量を越えて優先指定情報が付された場合には、使用順に常駐化バッファ8にロードし、常駐化バッファ8の容量を越えた分のプログラムは外部ファイル1に格納しておき、一般のプログラムと同じように外部ファイル1から直接ユーザ・プログラムエリア5にロードすることになる。

ルート・モジュール6からのオーバレイ・プロ

グラム・ロード要求の度ごとに、プログラム・ローダ4が優先指定情報の有無を判定して、優先指定有りのときはインデックス部を参照し、該当するプログラム・モジュール9があればプログラム常駐化バッファ8からユーザ・プログラム・エリア5に該当プログラム・モジュール9をロードしまた該当プログラム・モジュール9が無ければ、外部ファイル1からプログラム常駐化バッファ8にロードし、ロード済みであることをインデックス部7に登録した後、これをユーザ・プログラム・エリア5にロードする。これにより、プログラム常駐化バッファ8の容量制限により常駐化できるオーバレイ・プログラム・モジュールの数が限られていても、使用頻度の高いプログラム・モジュール、つまり優先度の高いモジュールを常駐化することができ、それによつてメモリ容量を最大限に有効利用できるとともに、ロード時間を短縮することができる。

第2図(a)(b)は、第1図の処理フローチャートである。

プログラム・モジュール9には、使用頻度の高いものと、低いものとがあり、これはユーザ・プログラム作成時にオーバレイモジュール化した時点で判断することができる。したがつて、ルート・モジュール6は、あらかじめどのプログラム・モジュール9を優先指定の対象とするかを知っている。第2図(a)はルート・モジュール6による処理フローチャートであり、第2図(b)はプログラム・ローダ4によるロード処理フローチャートである。

オペレータからの起動によりロードされたルート・モジュール6は、業務処理を実行する際に、必要なオーバレイ・プログラム・モジュール9をユーザ・プログラム・エリア5にロードする。このとき、第2図(a)に示すように、先ず使用頻度の高いプログラム・モジュール9に対して優先指定が必要か否かを判断し(ステップ21)、必要であればプログラム・ローダ4を起動させるための起動情報に優先コードをセットし(ステップ22)、プログラム・ローダ4を起動させる(ステップ23)。使用頻度の低いプログラム・モジュール9に対し

ては優先指定を行わず、プログラム・ローダ4を起動する(ステップ21, 23)。

第2図(b)の④は常駐化エリア8のインデックス部7を参照する処理であり、⑤は常駐化エリア8が満杯のときのロード処理、または優先指定のないときの処理であり、⑥は常駐化エリア8が満杯でなく、空きがあるときのロード処理であり、⑦は常駐化エリア8に該当プログラムが常駐化していたときのロード処理である。

プログラム・ローダ4は、優先指定の有無を判断し、優先指定有の場合には、インデックス部7をサーチし、ロード要求のあったプログラム・モジュール9のプログラム・モジュール名称がインデックス部7に存在するか否かを調べる(ステップ25, 26, 27)。インデックス部7に存在しない場合には、そのインデックス部7の内容から常駐化エリア(ここでは、第2エリア)8が満杯であるか否かを調べる(ステップ29)。満杯でなければ、外部ファイル1から該当プログラム・モジュール9を先ず常駐化バッファ(第2エリア)8

にロードし、インデックス部7にそのプログラム・モジュール名称を登録する。(ステップ30, 31)。次に、常駐化バッファ(第2エリア)8からユーザ・プログラム・エリア(第1エリア)5に該当プログラム・モジュール9を移動して、処理を終了する(ステップ32)。

また、常駐化バッファ8が満杯のときには、外部ファイル1より直接プログラム・エリア(第1エリア)5に該当プログラム・モジュール9をロードする(ステップ33)。

また、常駐化バッファ8に、該当プログラム・モジュール9が存在する場合には、単に常駐化バッファ(第2エリア)8からプログラム・エリア(第1エリア)5に移すだけでロードできる(ステップ28)。

この処理により、繰り返し同一のプログラム・モジュール9がルート・モジュール5によりロード要求された場合、インデックス部7に既に登録されているため、外部ファイル1からロードすることなく、常駐化バッファ8からユーザ・プログ

ラム・エリア5に移動するだけでよい。したがって、ロード時間は短縮される。

また、優先指定なしでロード要求されたプログラム・モジュール9は、外部ファイル1から直接ユーザ・プログラム・エリア5にロードされる(ステップ33)。

このように、本実施例では、優先指定のあるプログラム・モジュール9は、最初のロード要求時のみ、外部ファイル1からロードされるが、2回目以降はメモリ間の移動のみでロード処理を遂行することができるので、ロード時間が短縮され、かつ有限であるプログラム常駐化バッファ8を最大限に有効利用することができる。

(発明の効果)

以上説明したように、本発明によれば、内部メモリにユーザ・プログラム・エリアとは別個のプログラム常駐化バッファ・エリアを設け、この常駐化バッファ・エリアに常駐している優先指定付加のプログラムをユーザ・プログラム・エリアに移すのみでロードを実施できるので、使用頻度の

高いプログラムのロード時間を短縮でき、かつ有限のメモリ・エリアを有効に利用することができる。

4. 図面の簡単な説明

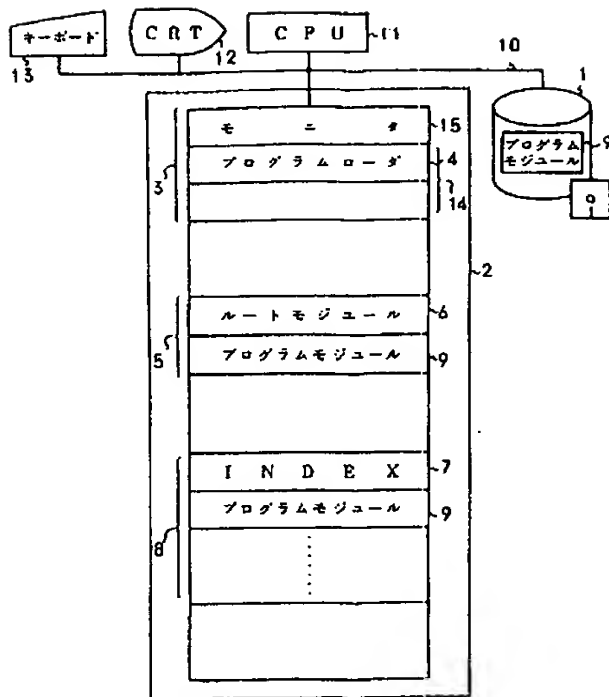
第1図は本発明の一実施例を示すシステムのブロック図、第2図は第1図の処理フローチャートである。

1: 外部ファイル、3: OS格納エリア、4: プログラム・ローダ、5: ユーザ・プログラム・エリア、6: ルート・モジュール、7: インデックス部、8: プログラム常駐化バッファ、9: プログラム・モジュール、14: OSモジュール群、15: メモリ。

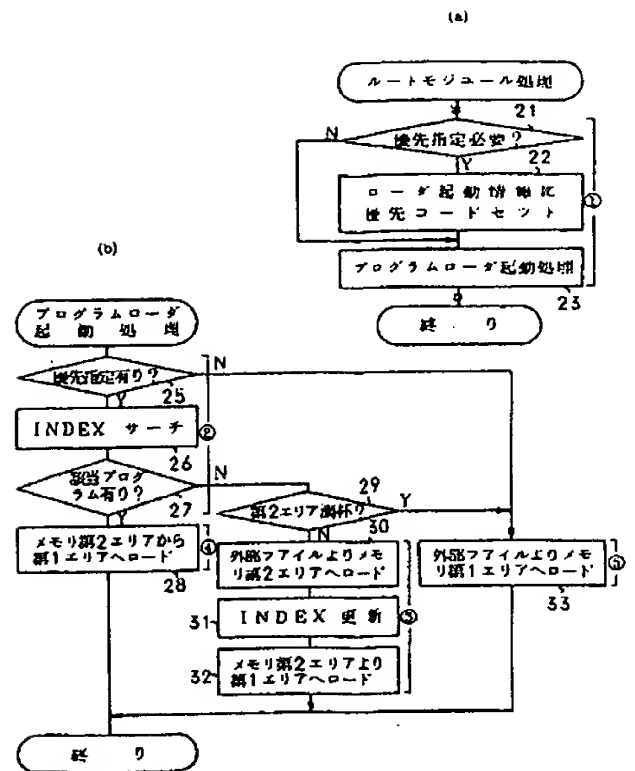
代理人 弁理士 高橋明夫

BEST AVAILABLE COPY

第 1 题



2



BEST AVAILABLE COPY